LEARNING PATH RECOMMENDATION -A SYSTEMATIC MAPPING

Guilherme Medeiros Machado, Anne Boyer

Projet « PLANETE », PO FEDER 2014-2020 / OS18.2
Action N° : 3
Delivrable N° : 1 – ÉTAT DE L'ART
Nancy 09/02/2021

LEARNING PATH RECOMMENDATION – A SYSTEMATIC MAPPING

Guilherme Medeiros Machado, Anne Boyer
Université de Lorraine, LORIA – Équipe BIRD
Nancy - France

1. INTRODUCTION

PLANETE is the acronym for "Projet Lorrain Ambition Numérique En Territoires pour l'Education", it is a French project focused on answer some broad challenges faced by the use of digital learning in Nancy-Metz academy. Such challenges are related to the proximity between academy and classrooms, the regional equity, the e-education, and the promotion and uses of enriched pedagogical practices. The project is divided in three axes which LUNE is one of them. LUNE is a platform that recommends pedagogical content, applications and digital tools to be used in learning scenarios. The platform aims are fourfold:

- To give visibility and access to digital resources and good practices developed in the Nancy-Metz academy on Lorraine region.
- Propose educational resources integrating (through the use of such resources) immersive simulation situations.

- Diversify, enrich and share teaching and training methods through the access of resources, digital tools and testimonials of uses and experiments around the digital world.
- To attend the whole Lorraine region, while promoting local specificities.

This platform will be a support and a swarming tool to be adapted to the digital culture of teachers. The core of the platform is a recommender engine which will suggest learning paths based on the users' interests. The resources' sources are the Nancy-Metz academy websites which will be aggregated and displayed in LUNE.

By promoting the Nancy-Metz academy learning content we expect to foster good teaching practices from the academy to the teachers in Lorraine region. The teachers who want to update their knowledge about a subject, or to study a totally new subject will benefit from LUNE. The use of e-learning (learning through computer tools) is strategic to facilitate such knowledge transfer because learning material generated by the academy can spread rapidly while generating very low costs.

The pandemic context has also stressed this project relevance. The impossibility to go to a physical classroom to take a course made the use of e-learning no longer complementary but mandatory. This sudden changing of teaching paradigm has generated a huge amount of good digital learning resources but not always providing the proper tools to handle such resources.

The creation of learning paths (an ordered sequence of learning resources to achieve a specific learning goal) by the teachers is one of the ways to guide the students through the overwhelmingly number of resources available on internet. Such construction of learning paths aims to avoid the students' cognitive overload and consequently dropout. However, such learning-paths are often built by the teacher without knowing the students learning styles, backgrounds, preferences, i.e., the student's profile.

It is known since the end of the 1990's, when the use of e-learning started widespread, that when dealing with multiple students' profiles in an e-learning environment "one-size-fits-all" solutions (provide the same set of learning resources, to different students) are not effective [1]. Different students should receive different resources, that should be adapted to their profiles. In the context of informal learning, where students are not engaged in a formal course with evaluation and grades, the students are known as lifelong learners. LUNE also to help such users, and in such contexts the challenges are even bigger because often in such scenarios the user is not an expert in the domain and could be rapidly overwhelmed if presented to a bunch of resources without a logical order or grouped under a consistent category. That is why is so important to create a platform like LUNE, which will help teachers, students, and lifelong learners by searching, filtering, and recommending learning paths adapted to each unique profile.

The challenges when proposing a platform like LUNE are found in the personal and technological aspects. In personal aspects we should assure this platform will help the teachers and students but will not replace the teacher. One of the ways to motivate the use and acceptance of LUNE is by ensuring transparency. It is highly unlikely to teachers accept a suggestion to ameliorate a standard learning path that they are used to use without explaining why such path is so important and why it is better to adapt that way to fulfill the current students' needs and profiles. By explaining the recommendations, the users are reassured and can engage easily while we reaffirm the role we want to play with LUNE serving as a companion platform where teachers, students and informal learners can meet to update their knowledge using the Nancy-Metz academy published leaning objects.

Other personal challenge is related to the trust the users will put in the recommendations. Trust is a judgement of unquestionable utility [2] and even if the learning resources presented are created by human specialists that can be known by the teachers in the region, the generation of the learning paths are created by an algorithm they do not know and are not always ready to trust. So, we also need to ask how to create trust between the user and the platform? How to measure this trust level? How to assess if by trusting the users are more engaged and effectively using the platform?

In a technological view of the challenges, we can ask questions about how to model a user knowledge and the learning resources context? How to match the users' learning objectives to a successful learning path? How to assess if the recommendations really helped the users in the achievement of their objectives? How to treat users from many different contexts, ages and backgrounds? How to aggregate multiple data sources with multiple rights of access? Even further, how to assess the effectiveness of LUNE in providing knowledge transfer?

To answer these and other questions we propose this systematic mapping of the literature. During the rest of this text, we will try to answer such questions while addressing how some of these challenges are treated in the literature. The following sections will present the methodology of research, the results, and the conclusions of our mapping.

2. Conceptual Foundation

Recommender systems are software to suggest relevant items to their users. They have their roots in the beginning of the 1990's. They were firstly proposed to deal with the problem of cognitive overload many users were experiencing when managing the ever-increasing amount of information made available through personal computers. The first proposition found in the literature is Tapestry [3], which was a system proposed to manage incoming e-mails of a company.

The field have greatly evolved from such epoch, and nowadays we can find two main strategies for providing recommendation, either by using the intrinsic characteristics of an item to match it with the user preferences, this is known as content-based filtering; or by matching users by their items consumption and recommending new items the similar users have chosen and liked, such strategy is known as collaborative filtering [4].

These two paradigms of recommendation, content-based and collaborative filtering are better explained in the following subsections.

Content-based

Content-based recommendation is an intuitive way of recommending items, that relies basically on a description of the item (known as content), and a user profile containing the user past interactions with the items. One of the drawbacks of such technique is the need of having items well described. For instance, to recommend a movie using this technique it is necessary to provide description like, the genre, the list of actors, the director, a brief description of the movie, between other technical features and characteristics of such item. Other challenge to implementation of content-based recommendation is the discovery of qualitative features, that refers to the reasons someone has liked an item. The qualitative features present an even bigger challenge than the descriptive content, that commonly are provided by manufacturers about the items. Qualitative features in preference domains reflects the reasons why a user has liked an item; this reason sometimes is not related to intrinsic features or characteristics but could be instead related to an exterior design of a product.

Despite such drawbacks, the content-based recommendation presents some advantages facing other recommendation strategies. The first one, is the need to rely only on the item content and a user profile that reflects interactions with the items. This makes the content-based strategy able to provide suggestions of items even if the whole community of users comes down to a single user. Content-based is also able to provide recommendations even with little interaction of the user with the items. So, it is not affected by the classic *cold-start* problem (the difficult to provide good recommendations to new users or items) that affect the collaborative strategies. Such strategy is most fitted to recommendation in domains where the items are texts, or news, or have their features presented in texts. Such recommendable items will be referred as documents.

This ability to treat texts comes from some techniques and algorithms inherited from the field of Information Retrieval, that has played an important role in providing knowledge also to the learning of the user profile in content-based recommendation.

TF-IDF AND THE VECTOR SPACE MODEL

One of the most popular strategies to derive item features to be used in content-based recommendations is the vector space model and TF-IDF. Consider a document set that has its content represented by vectors containing all the words presented in the set and each time a word is found in a document the position is set with 1, otherwise with 0. In this simple model, the requirement to do a recommendation is to build a user vector with the words the user has interest in set with 1 and match the user vector with the document vector. This naïve approach, however, does not take in consideration the case the longer is the document, the higher is the probability to be recommended. So, it is necessary some strategy to avoid longer documents to be more recommended.

TF-IDF (*term frequency-inverse document frequency*) comes as a proposition to overcome this problem. Instead of describing the documents using Boolean keywords, they are encoded as TF-IDF vectors in a multidimensional Euclidean space. Each keyword now is obtained by the product of two sub-measures; term frequency, and inverse document frequency.

Term frequency (TF) is the number of times a term 'i' appears in a document 'j', this has the assumption the more frequent is a term the more important it is. However, to prevent longer documents from getting a higher relevance is necessary normalize this measure. So, it is calculated as:

$$TF(i,j) = \frac{freq(i,j)}{maxOthers(i,j)} \qquad (1)$$

Where freq(i,j) is the number of times the term 'i' appears in the document 'j' and maxOthers(i,j) represent the number of the maximum frequency between all the other terms present in the document.

Inverse document frequency (IDF) is a measure that aims to reduce the importance of a term that appears frequently in the set of documents and consequently is not discriminative of the target document. IDF is calculated as:

$$IDF(i) = log\frac{N}{n(i)} \qquad (2)$$

Where *N* is the number of all recommendable documents and *n*(*i*) is the number of documents of N which *i* is present. The product of TF and IDF gives the TF-IDF metric used to describe the documents using vectors.

$$\textit{TF-IDF}(i,j) = TF(i,j) * IDF(i) \qquad (3)$$

After having the items' profile vector computed we can apply a cosine similarity to get a rank of other similar items. A user profile can also be built using the TF-IDF vector space, this allows the verification of similarities between user and items. Following such strategy the item selection problem in content-based filtering can be described as “recommend items that are similar to those the user liked in the past” [4].

Collaborative Filtering

Collaborative filtering approaches rely on user’s past behavior or the opinions of an existing user community to provide recommendations. These type of recommender systems are widespread in many industry applications such as online retail sites, media streaming services and educational platforms. The collaborative approach is well known and form many years its algorithms and techniques have been studied. The pure collaborative approach takes a matrix of “user x item” ratings as input and produce two outputs: (i) a prediction of ratings the user would give to a certain item; and (ii) a ranking of the top-k most likeable items. The ranking traditionally does not contain items the user has selected previously. However, recent research efforts have proposed a mixt ranking of familiar and new items in the ranking to promote user-system fidelity [5].

The collaborative approaches can be divided in neighborhood-based and model-based approaches. There are two possibilities to recommend using a neighborhood-based strategy. The first is to take the similar users' ratings as basis to estimate the current user rating to an item; this method is called user-based. The second is to consider the ratings the current user gave to similar items, to estimate the current item rating, two items are considered similar if their ratings between the users of the system are similar, and this method is called item-based.

Model-based approaches have their recommendations generated by looking to a portion (training set) of the user’ ratings and learning a behavior model. Such model is then used to retrieve the user ratings for new items. One characteristic of such approaches is they are more accurate than neighborhood-based ones. One of the most common strategies in model-based collaborative filtering is the factorization matrix.

In what concerns the challenge of providing trustworthy recommendations, we highlight the work of Massa and Avesani [6]. They were one of the first authors to propose a way to measure the trust between the users of a recommender system and use this metric to quantify the similarity between users. The authors model the users inside a trust network where it is possible to find the most trusted users. Such strategy can be useful to LUNE scenario, if we are able to provide recommendations with a trust bias (i.e., calibrate the recommendations to follow each individual most trusted peers) we believe this strategy can also increase user engagement.

Hybrid Methods

One popular way to provide recommendation is by combining the collaborative-filtering and content-based algorithms, such combination generates a hybrid method. The goal of a hybrid method is to take advantage from the strengths of each recommendation strategy while limiting their drawbacks. To illustrate a hybrid method, we can think about a recommender system that uses matrix-factorization to generate recommendations when there is enough data about an item or a user, but it will use a TF-IDF algorithm to generate recommendations to new user and items. In this example the goal of the hybrid method is to avoid the famous *coldstart* problem, where the system cannot infer any recommendations to users or using items that it has not gathered sufficient information. *Coldstart* is particularly present in collaborative-filtering algorithms, however the use of a TF-IDF algorithm to recommend to new users while the system gathers enough data to switch to a matrix-factorization can solve this problem in this particular scenario. The problem of *coldstart* is then solved, but it does not mean the system will be flawless. A content-based strategy is known for generating recommendations more homogeneous recommendations than collaborative filtering, this could discourage the user in proceeding with the system utilization, since the recommendations will present more of the same.

Some of the most common ways to combine recommender strategies are classified in [7], they are:

- Weighted: where the scores of many recommendation techniques are combined to produce a single recommendation.
- Switching: where the system switches of techniques depending on the situation (as described in our example).
- Mixed: different recommendations from different algorithms and techniques are presented at the same time.
- Feature combination: where features from different data sources are combined into the same recommendation algorithm.
- Cascade: one recommendation engine refines the recommendations given by another one.
- Feature augmentation: the output from one technique is used as input features to another technique.
- Meta-level: the model learned by one recommender is used as input to another.

Independently of the technique used to combine the algorithms, the hybrid methods became really popular because of their efficacy in providing good recommendations. For LUNE context, we think a hybrid approach could help solve the challenge of multiple data sources, we could use a "Feature combination" strategy to extract recommendations from the many sources of the Nancy-Metz academy.

Learning-path Recommendation

We want to show in this subsection that the task of recommending a learning path is different of a traditional recommendation task. The most common output of a traditional recommender system is a list of items ranked by relevance, i.e. the first item fits better the user preferences and interests than the second item, the second item fits better than the third and so on. This ranked list is well suited to many situations and domains of recommendation; however, when we treat the recommendations of learning objects used to accomplish a learning goal, we cannot provide a simple list of ordered objects.

The problem when using a simple ordered list to present learning recommendations is that a bag of "disorganized" learning materials will disorient and overload the user [8]. We say disorganized because the learning materials follow an order other than the one suitable to accomplish the learning goal.

We need to present the recommendations as a sequence where we consider the dependency between the items that form such sequence. For instance, if the user wants to learn about the homo sapiens evolution, we need to present learning objects that talk about the behavioral modernity (happened around 50 thousand years back) before talking about the industrial revolution. However, such sequence of learning items must also consider the user profile (e.g., their preferences, competences, learning styles, previous knowledge…), otherwise users with different backgrounds will receive the same learning path. In our example a user that is specialist in evolution should receive a path of learning object much more detailed than a user that wants to get in touch for the first time with the subject.

Another challenge concerning the recommendations of such sequence is how to capture all the important aspects of the user profile to a specific scenario. For LUNE project we need to think about the data that is available to be processed, how to recognize the user learning style or previous knowledge through the analysis of simple traces left by the use of e-learning platforms? Even further, how can we model the dependency between objects? Returning to our example, how can I recommend a learning object about the behavioral modernity before the learning object that will treat about the industrial revolution? Some of such questions will be answered by the analysis of the papers in this mapping.

Once we modeled the users and the items, we also need to think about the best way to match their profile with the learning sequences. This is the task that will be performed by the recommender system. But since we are treating of a such specific kind of recommendation, we also need to look to the literature to find the best techniques. By looking to our example, we could think the modeling problem in computational matters if we consider each learning object as a node of a directed graph[1] and the edges between such nodes would represent a pre-requirement relation. Once we modeled the domain, we could think about a novel algorithm to find a path in such graph that will maximize the probability to reach the learning objective for a specific user profile. Such strategy is one of the most used when recommending learning paths, as we will show later on this document. We also need to think about the teachers' freedom of choice, the learning paths should not be intrusive and the teacher will be free to choose between using the recommendations or doing their own learning path manually in LUNE.

Evaluation of Recommendations

When looking to the recommender systems evaluation 80% of all the papers published in the biggest conference of recommender systems (RecSys) treated the problem to enhance their rating prediction accuracy, while only 20% tried to enhance aspects other than accuracy [9]. This is statistic is especially important to think about the advances made in the field and how such advances are important to our research context. LUNE is a platform that will provide recommendations of learning objects and by doing that will expect to transfer knowledge from the academy to teachers and users interested in the platform. Looking to such objective we can know that LUNE will need to develop a way to evaluate the impact the use of the platform did in the learning. The assessment of such learning is an indirect measure, because it cannot be extracted directly from the recommendation data. We need to ask the user about the learned content.

Other concern of LUNE when evaluating is how to stimulate the users' engagement with the platform. The goal is to make the platform a service that is often used. One of the ways to stimulate the user engagement is to make them trust in the presented recommendations. How to stimulate the user trust if the mechanism of recommendation acts as a black box? We need to explain the users the reasons they are receiving such recommendations. The assessment of the recommendations through a fitness too the user goal but also through the capability of explaining is one of the goals here.

The assessment of the learning path fitness can be performed by using an optimization function, as well as, demanding the user opinion about the recommended path. Such user opinion will then feed the mechanism of recommendation to provide items better suited to each user profile. In what concerns the assessment of the explanations we can use metrics as Mean Explanations Precision (MEP) and Mean Explanations Recall (MER), as well as the user opinion about the explanations.

---

[1] In mathematics, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices (also called nodes or points) which are connected by edges (also called links or lines). A distinction is made between undirected graphs, where edges link two vertices symmetrically, and directed graphs, where edges link two vertices asymmetrically. (Source : https://en.wikipedia.org/wiki/Graph_theory)

THE LUNE CONTEXT

Once we are aware of the different techniques to provide recommendation and the challenges of providing recommendation in learning scenarios, we ask ourselves: How to propose a recommendation engine that will suggest learning paths to facilitate the knowledge transfer in Lorraine region?

The specificities of the task demand an effective recommendation engine that will be accurate, but also transparent, and trustworthy. The biggest challenge is to maximize the probabilities of the teachers to engage with the platform. It should assure its role as a companion to learning and not a replacement of the teacher. So, it should be interesting to provide a way to teachers give their feedback about the recommendations too. The next section will present the methodology we used to search the literature about solutions that are similar to LUNE, their main characteristics and what could we learn an apply in LUNE context.

3. THE METHODOLOGY

A systematic mapping is defined by Petersen et al. [10] as a "method to build a classification scheme and structure a software engineering field of interest". This is a rigorous method to perform literature reviews, and despite being proposed to be used in the software engineering field it is now applied to many of the computer science fields. The method has three main phases: i) Planning, ii) Conducting, and iii) Reporting, each of which is detailed in the next subsections. Some of the advantages of performing a systematic mapping to review the literature are the reproducibility of the results and the decreasing of personal bias. A systematic method can also guarantee the explainability of each decision took during the mapping. The planning of each action performed during the conducting phase will allow to explain each inclusion and exclusion criteria used to a retrieved paper. Clear constraints defined during the planning will also allow to explain the criteria used to form the search string, the quality assessment questions, and the information extraction.

PLANNING

The main objective of this mapping, as already explained in the introduction, is to **search the literature for learning path recommendation approaches and tools to start the development of LUNE platform**. The first task then is to define the planning of such review, such planning will define the steps that will guide all the rest of the mapping. The first task is to define the **research questions** (RQ), which will cover specific objectives and will guide the definition of a **search string** (second task) that will be submitted to some selected **search engines** (third task). To this paper, we have defined the following research questions:

RQ1. What are the algorithms used to perform Learning-Path Recommendation (LPR)?
RQ2. What are the tools that can perform sequential or Learning-path recommendation?
RQ3. What are the differences between LUNE and the other existing tools?
RQ4. How such algorithms and tools are evaluated?

While the main objective defines "what" we search, these five RQ split the main objective into more specific ones asking "what" and "how" things happen in the domain of Learning-Path

Recommendation. The RQ1 is defined to understand the way the recommendations are performed in this domain; such question is related to the objective of discovering the state-of-the-art (the highest level of development in a scientific field at a particular time) based on the comparison of algorithms used to recommend. It is also related to two different computational challenges cited before: "how to model a user knowledge and the learning resources context?" and "how to match the users' learning objectives to a successful learning path?". The RQ2 tries elucidating the state-of-the-art by looking to the developed tools and projects that perform activities similar to the ones in LUNE. The RQ" is a consequence of RQ2, because once known the other projects what can be used in LUNE and what LUNE is different from these tools. Finally, the RQ4 tries to elucidate how the propositions to learning path recommendation discovered in RQ1 and RQ2 are evaluated, such question is also related to the computational challenge of "How to assess if the recommendations really helped the users in the achievement of their objectives?".

Once we have our RQ we can translate it into a query to be submitted in many academic search engines. By following the general rules of the engines our search string is defined as:

("Professors" OR "Students") AND ("Learning Analytics" OR "Learning Objects" OR "Learning-path" OR "Learning Recommendation" OR "TEL Recommendation" OR "Tools") AND ("Book Recommendation" OR "Sequential Recommendation" OR "Learning-path Recommendation" OR "Pattern Recommendation")

Such string shows the population of interest to this mapping (Professors, Students), the context (Learning Analytics, Learning-Objects, Learning-Path, Tools, etc.) and also the more specific interests of the mapping (LPR, Sequential Recommendation, etc.). This string is defined by using keywords and logic connectors to be submitted to six of the most popular academic search engines in computer science, they are:

- ACM Digital Library[2]
- Google Scholar[3]
- IEEE Xplore[4]
- ScienceDirect[5]
- Scopus[6]
- Springer Link[7]

Due to the huge number of results returned by Google Scholar engine (1480 papers) we analyzed the retrieved results to define a cut point. Since Google Scholar rank their results by relevance our criteria to define the cut point was to consider all the results of a page if at least one of the papers in such page present a title related to Learning-Path Recommendation. Knowing that each page of result present 10 papers and following the cut point criteria we stopped at the 5th page of results, i.e. we retrieved 50 papers, because the 6th page did not presented papers which the title was related to Learning-Path Recommendation. More details about the conduction are present of the mapping are presented in the next section.

Conducting

[2] https://dl.acm.org/about
[3] https://fr.wikipedia.org/wiki/Google_Scholar
[4] https://ieeexplore.ieee.org/Xplorehelp/overview-of-ieee-xplore/about-ieee-xplore
[5] https://www.elsevier.com/solutions/sciencedirect
[6] https://www.elsevier.com/solutions/scopus?dgcid=RN_AGCM_Sourced_300005030
[7] https://link.springer.com/

Once defined the search engines, we submitted the search string and collected all the results returned, with exception to scholar google where we used our cut point criteria. The Table 1 show the results after submitting the string (Imported Papers), the first screening (Accepted), and the quality check. Figure 1, shows the distribution of retrieved papers by each search engine, since Scopus indexed some of the other engines it has returned a considerable number of papers, but it also contributed to the generation of duplicated papers. For the SpringerLink, the high number of papers is explained by the worst search tool among all the other engines used in this review, the results returned was not as accurate as it was in the other engines.

TABLE 1 - THE NUMBER OF PAPERS RETRIEVED AFTER EACH FILTER

Source	Imported Papers	Accepted	Quality Check (>4.0)
ACM	28	1	0
Google Scholar (5 pages)	50	8	5
IEEE Xplore	6	1	0
Science@Direct	29	2	0
Scopus	220	29	16
Springer Link	234	8	5
Total	567	49	26

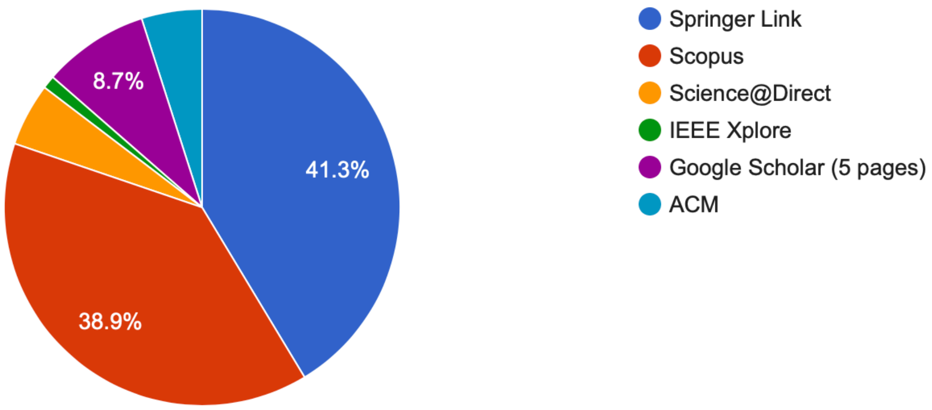

FIGURE 1 - THE RETRIEVED PAPERS BY SEARCH ENGINE

For the first screening we used a list of inclusion and exclusion criteria also defined during the planning phase. To include a paper into those which will pass through quality checking it should present one of the reasons presented in the inclusion criteria list, the exclusion followed the same logic. The papers' title, abstract, and keywords were enough to perform this screening. At the end of this screening the list of papers was narrow down from 567 to 49 items coming from the six search engines. The Figure 2 presents the reasons to exclude a paper during the screening, among the most common reasons are the fact that the paper was neither from recommendation nor from learning domain. This reason was particularly present among the papers retrieved by SpringerLink because their search tool presented problems in filtering non relevant results, the tool presented some papers from the nutrition or even medical domain. Other important reason that appeared in our screening is the fact that a paper belongs to the educational domain but is not related to recommendation in any ways, this was verified among the many papers retrieved from the psychological and educational domains but that were not related to computer science. Many secondary studies ("the summary, collation and/or synthesis of existing research"[8]) were retrieved too, but this review is interested only in primary studies, some other papers presented approaches to recommendation in the educational domain but were not related to

[8] https://en.wikipedia.org/wiki/Secondary_research

learning paths, and some other papers presented recommendation approaches but not applied to learning.

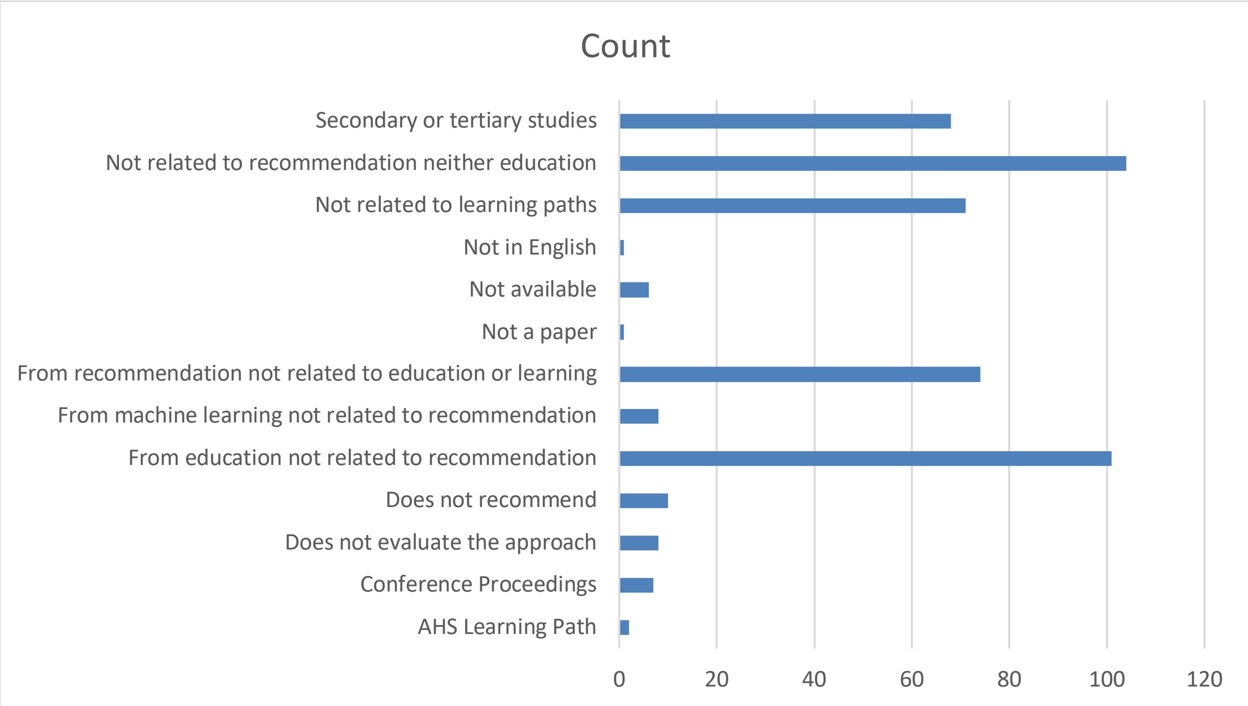

FIGURE 2 - THE EXCLUSION CRITERIA STATISTICS

In Figure 3 we present the most frequent reasons that lead to the inclusion of a paper into the list for quality checking. The most important reason is the paper present an algorithm or a tool for recommendation o learning paths. The second reason is the paper present and architecture, a formal model, or other description of its contribution for learning path recommendation, these papers were grouped under the category “Present a primary study”. One of the returned papers were a Google patent for Learning Path recommendation, it has passed the first screening, but it was rejected during the quality checking for not present a detailed and reproductible description of their contribution.

The Figure 4 shows the distribution of papers accepted and rejected after the first screening grouped by search engine, while the Figure 5 shows the same statistics but grouped by year. The year information shows an increase interest of the scientific community in the subject of learning path recommendation. We can see that the number of accepted papers has increased specially in the last three years, this is also confirmed by Figure 6, which shows a timeline of increasing number of papers treating the subject of LPRS. The complete list of papers with their status of accepted or rejected can be found here: https://1drv.ms/x/s!AugGORD5_DZWgewAnDhBTlRkKh5q3Q?e=YbRnwO

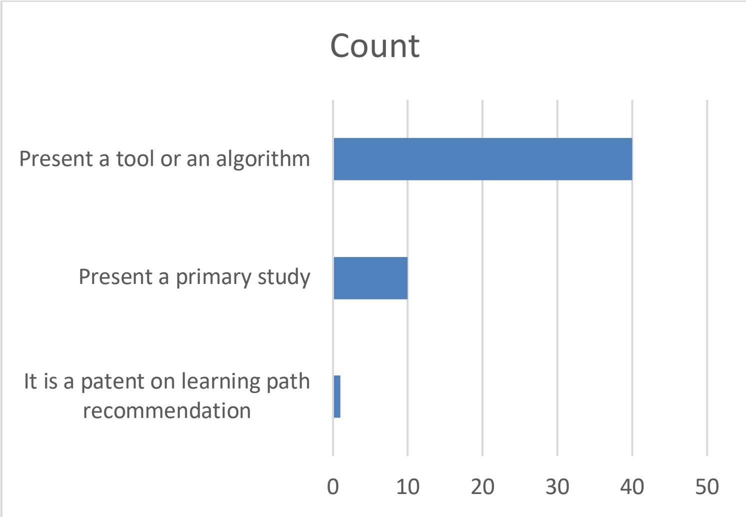

FIGURE 3- THE INCLUSION CRITERIA STATISTICS

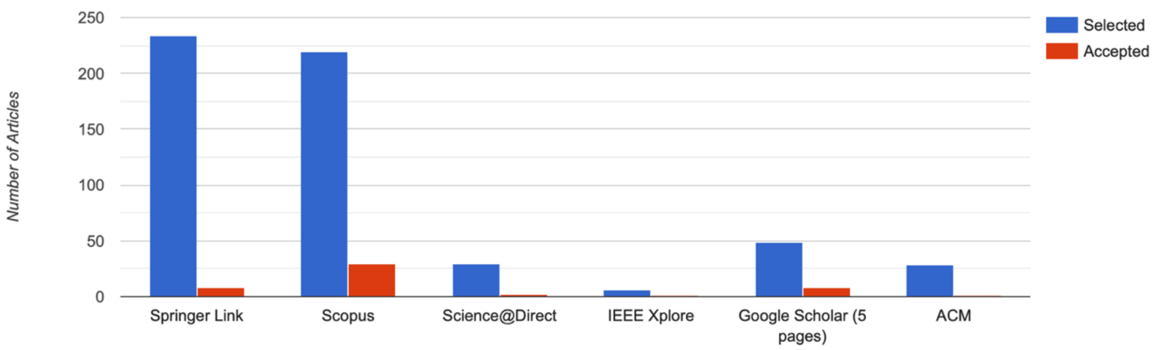

FIGURE 4 - THE STATISTICS AFTER THE FIRST SCREENING GROUPPED BY SEARCH ENGINE

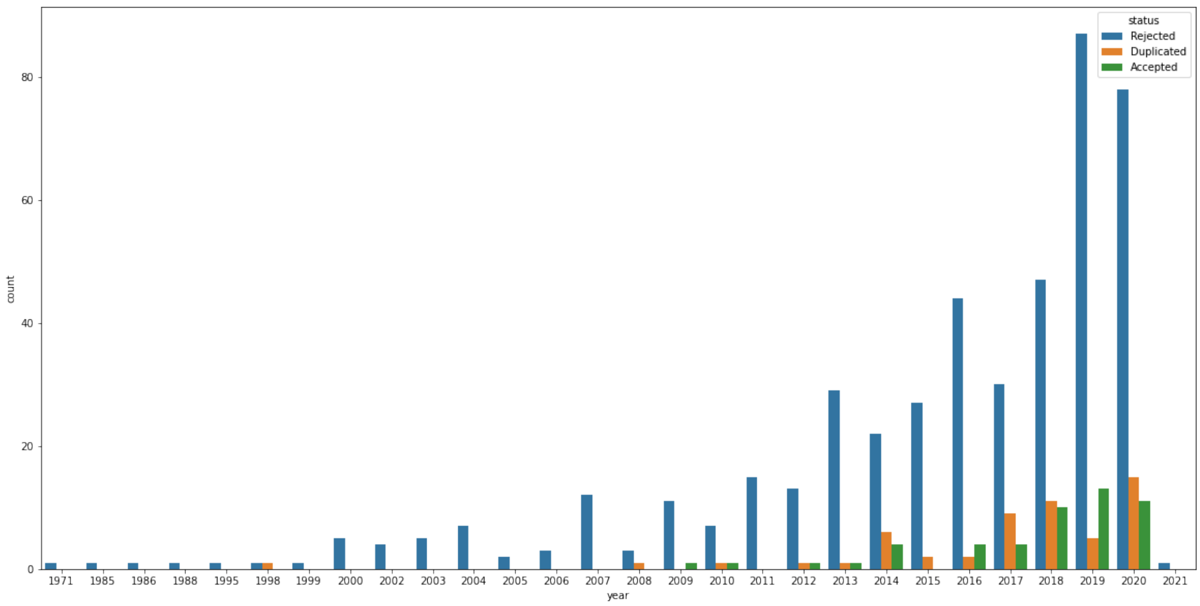

FIGURE 5 - THE STATISTICS AFTER THE FIRST SCREENING GROUPPED BY YEAR

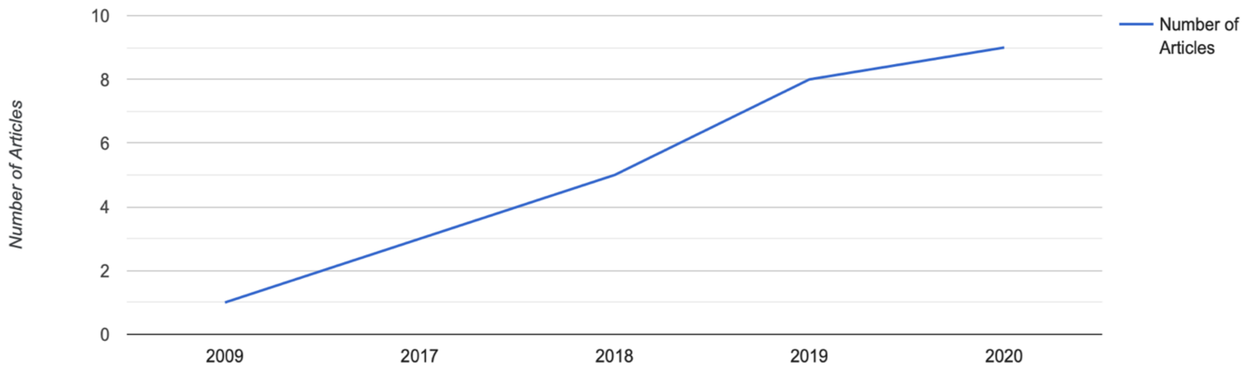

FIGURE 6 - THE NUMBER OF ACCEPTED PAPERS BY YEAR

After the papers were screened by subject a quality check was also performed. This quality check was conducted by answering a list of questions looking to each paper's content. Each paper scored 1 point if the answer to the question was 'Yes'; 0.5 points if answered 'Partially'; and 0 points if the answer was 'No'. The list of the questions is:

1. Was this paper published in the last five years?
2. The paper was published in a journal or conference specialized in education and computer science?
3. The problem and the solution are clearly described?
4. The paper presents a tool or is related to a bigger project?
5. Is their contribution related to learning-path recommendation?
6. Is their contribution compared to another in the literature? If yes, is it better?
7. Does this paper present an evaluation with quantitative and/or qualitative metrics?

The question 1 is important because we want to define the state-of-the-art in the field of learning path recommendation, so the more recent the paper is, the more relevant it will be to this review. Such recency will allow us to depict the frontier of the knowledge in this field and advance a little beyond such frontier with LUNE. The question 2 is important because we want to give more relevance to papers that were reviewed by computer scientists but also to educators, it is important to have a multidisciplinary view of the contributions in this field. The question 3 is asked to give more points to papers that are well written and well structured. The question 4 is directly related to the main objective of this mapping that is to find related tools and projects to LUNE. The question 5 wants to emphasize papers which main contribution was the development of a learning path recommender system, such question helped us to give less points to papers that used a learning path recommender but which the main contribution was related to other subject like the paper. The question 6 wants to give more points to papers we can replicate the results. Finally, the question 7 wants to assure the paper contribution was properly evaluated.

It is clear that the answer to such questions (with exception to the first two) demanded a deeper analysis of the 49 papers. Since the maximum rating a paper could achieve here is 7, and because the first two questions could not guarantee the inner quality of a paper, but they are of the interest for this survey, we defined as the cut point a minimal rating of 4.5. This minimal rating will guarantee that if the paper scores 2 points for the first two questions it will also need to cover at least half of the rest of quality questions. To perform such rating, we read the papers' sections that lead us to answer all the questions. At the end we had 26 selected papers to read and extract information to answer our RQ.

4. Results/Reporting

To answer the RQ defined in the beginning of this study and to characterize the contributions of each paper we proposed a form with questions to extract information from the papers selected after the quality checking. The questions presented in this form are thought to characterize the strategy developed to recommend Learning Paths, and how such propositions were evaluated. The form for data extraction is presented below. The questions are grouped into three groups related to the approach's nature, its quality, and its generality and reproducibility.

Approach's nature
• What recommendation strategy was used? • What kind of algorithm strategy was used? • What is the learning-path algorithm? • What is the approach output?
Approach's quality
• What kind of evaluation was performed? • What kind of learning metrics was used? • How many metrics are used to evaluate the approach? • The use of recommendation enhanced the learning?
Generality and reproducibility
• The tool or algorithm is publicly available? • The dataset is publicly available? • How many datasets used in the experiments?

Differently from the previous form, which was designed for quality check, this form can be answered by multiple checking boxes, single pre-defined answers, or free text. Seven of the eleven presented questions are presented as graphics and are discussed below. In what concerns the remaining four questions since they presented uniform or non-groupable results we discuss it briefly here. For the question about "recommendation enhanced the learning", all the papers presented positive results showing the use of a LPRS can significantly ameliorate the students score. The question about the "learning path algorithm" is not presented because each paper presented a different solution composed of generally of more than one algorithm; and to this combination they put a different name. So, each combination of algorithms of each paper had a different name. The last two questions presented very similar results where the majority of papers does not make their algorithms and datasets publicly available. Some of the papers presented a link to a dataset but such link directed us to a broken webpage. The only exceptions to these two questions are the paper of Nabizadeh et al. [11] which made available their contribution into a system born from a project of some of the authors, the system is easily found but it is not free. About the public available datasets none of the papers created a dataset and made it publicly available but three of the papers used public known datasets these papers are: Birjali et al. [12], Cai et al. [13], and Wang et al. [14].

Approach's Nature

The most common used recommendation strategy among the papers is content-based (Figure 7). This is caused because the approaches generally construct a pre-requirements chain between the items to be recommended, and to know such information we need to look to items content. To build the user profile the background and the acquirement of new knowledges caused by the use of the learning resources is taken into account too, what reinforces the need to look to the content. There is also an important number of papers that looks to the community of learners to mine a learning pattern and recommend a learning path to similar students, such papers are framed into the collaborative filtering ones.

The algorithm strategies are shown in Figure 8 and Figure 9, the first shows the results grouped by the nature of the used algorithm, showing that the use of machine learning and "Other non model based" are the two most popular manners to provide recommendation with learning paths.

Specifically, in Figure 9 we explode the machine learning category to show which specific algorithms are used. The other common strategy named "Other non-model based" refers to heuristic or greed-based strategies to find a path with the minimum cost into the graph of learning items, this path was then recommended to the student. To build such path the papers considered the learning items as the nodes of the graph while the arches represented the cost function determined by the paper objective, for instance the pre-requirements between objects, the demanded time, the increase of difficulty, the coverage of the subject, etc. Other important found strategy was the use of a genetic algorithm, the difference in such strategy is the generation of only one optimized learning path.

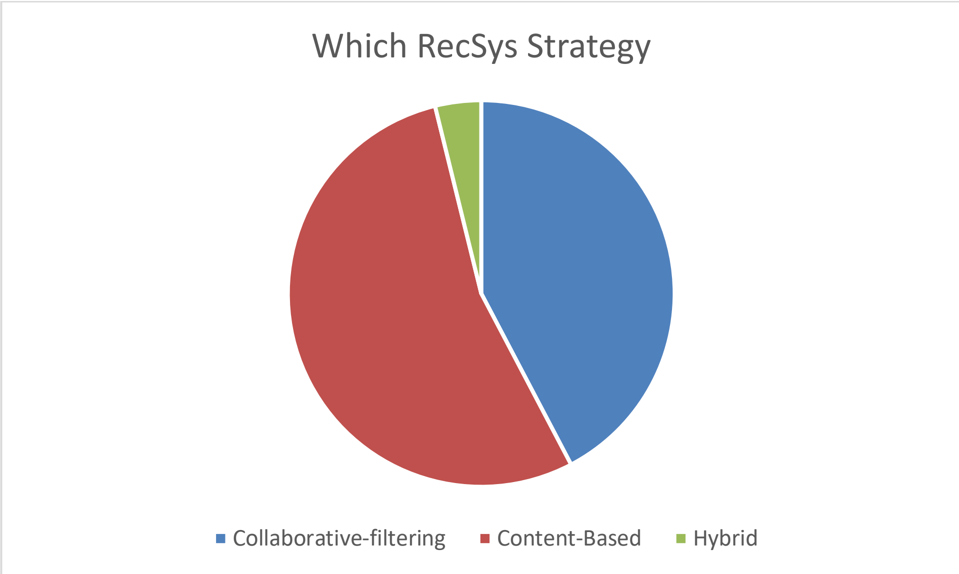

FIGURE 7 - THE DISTRIBUTION OF THE THREE RECOMMENDATION STRATEGIES

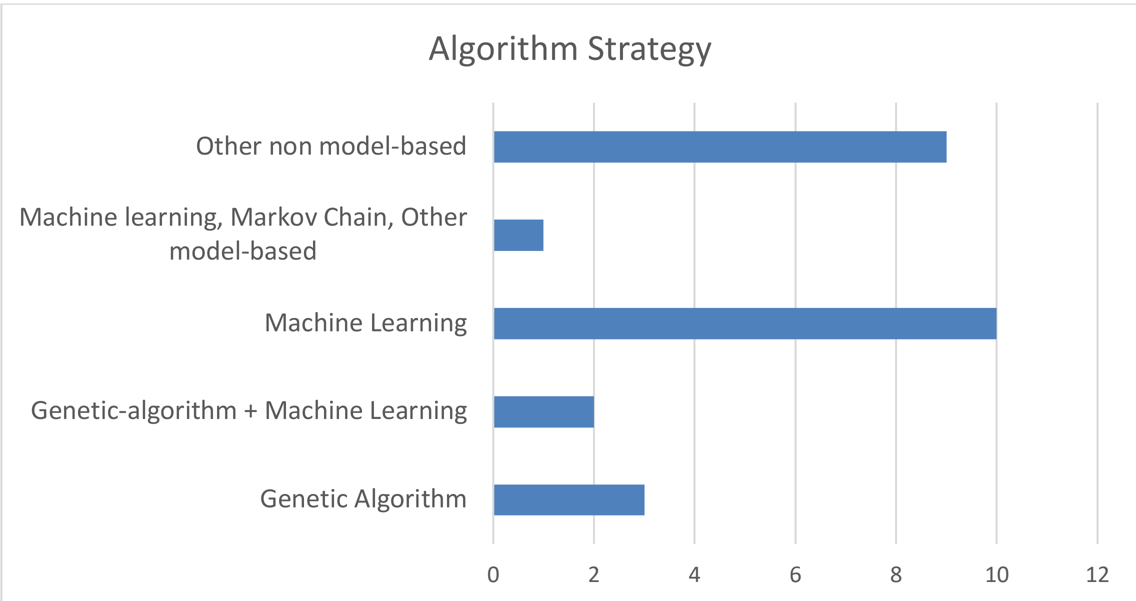

FIGURE 8 - ALGORITHM STRATEGIES GROUPPED BY NATURE

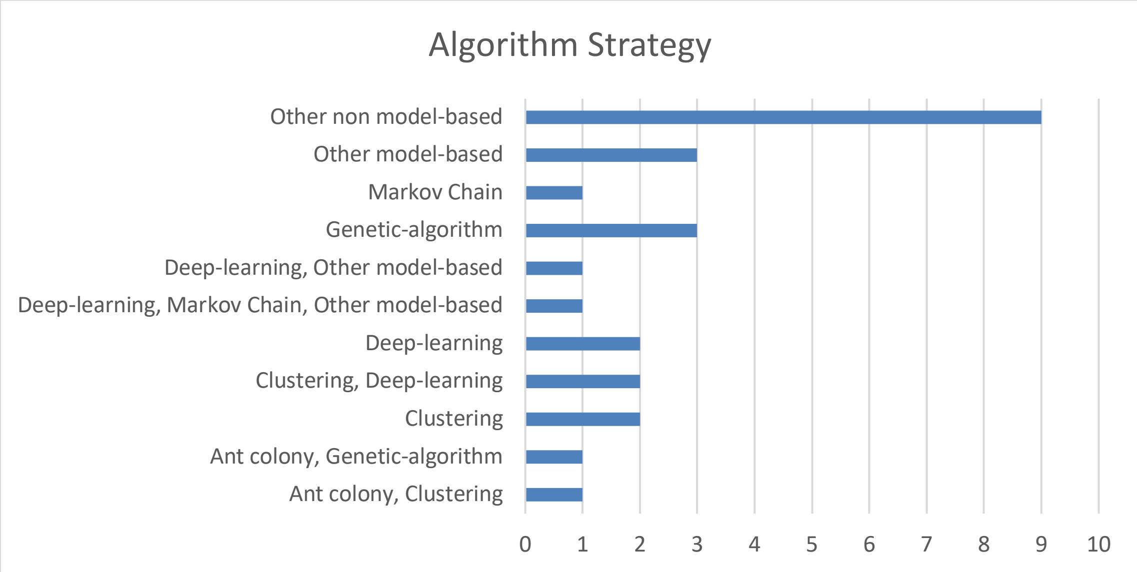

FIGURE 9 – SPECIFIC ALGORITHM STRATEGIES USED BY THE STUDIES

When looking to the approaches' outputs, the most common option is to propose a single optimized learning path. That means the learning path is generated and calibrated to select the path of resources that will accomplish the student goal (for instance learn about the industrial revolution) while minimizing or maximizing some other constraint (for instance the time demanded by the student to learn). Such constraint minimization/maximization is performed by optimizing an objective function, such optimization is performed by systematically selecting input values (in our case, the learning path) from a set of allowed ones and computing the output of such function (in our case, how much time will be demanded by the user). The output of such optimization is then a single recommended learning path. Such output limits the user choices but guarantees the user is not overwhelmed by the recommendations. For LUNE we need to find the best balance between the single output and the presentation of alternative recommendations, such balance will be determined by the potential users of the platform.

The other less popular types of outputs are listed in this paragraph. The first one is the presentation of "an undirected graph" (which is a graph with no relation of direction) what gives the opportunity to freely navigate among the learning resources, but it can also make the user easily get lost in achieving the learning goal. A "sequential list" is a sequence of items not organized in a graph neither optimized. "A list of next possible items" presents many options of items to be selected in the next step without the need to plan the whole path at once. "A list of many possible learning paths" is the option that shows in a graph or any other structure the many possible ways to go from a point A to B not necessarily optimized ones. Finally, a directed graph is a graph where the edges have a direction. The Figure 10 shows our finds.

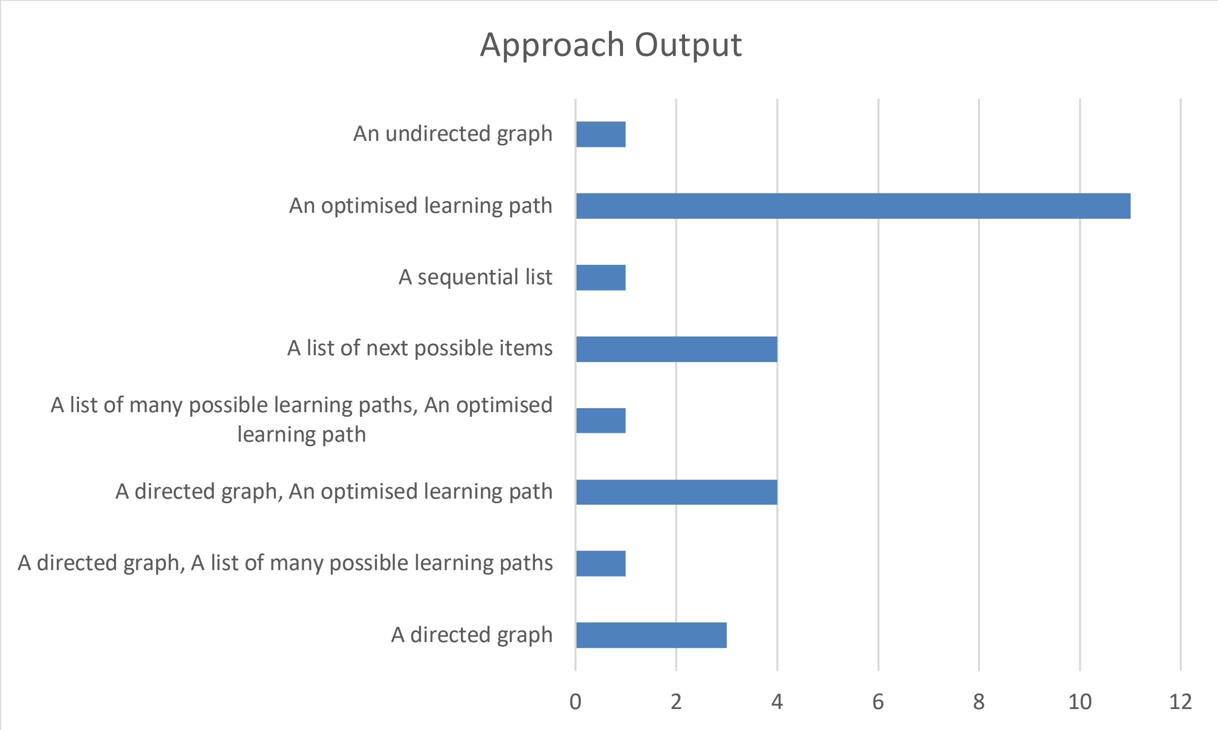

FIGURE 10 - HOW ARE THE LEARNING PATH PRESENTED OR DELIVERED BY THE APPROACH

APPROACH'S QUALITY

Regarding the type of evaluation performed (Figure 11), the two most popular types of evaluation found are the execution of online experiments (demands the user participation to collect their interactions with the proposed approach, it is generally web-based) and offline experiments (based on past users' interactions with the same or a similar approach) with real dataset. Concerning the datasets, we can distinguish the ones with real data, i.e., data generated by the user and the ones with data generated by a simulator to emulate the user behavior (simulated data). The use of a real dataset to perform offline experiments provide fast results with less effort and it is generally popular to evaluate machine learning approaches. However, in learning domains an offline experiment cannot assure that the use of the tool ameliorates the students learning because their learning is often measured using the score in a final exam, which is an undirect metric that can be influenced by the recommendation tool but not only by it. Common metrics to perform an offline evaluation of a recommender system are the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Precision, and Recall, that measures how accurate our recommender is predicting the user preferences. Other common metrics are Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (nDCG), that measures how accurate are the ranking of items presented to the user.

One common option to assess the enhancement of the learning then is the execution of online experiments. In an online experiment the learning performance is compared by splitting the students in two groups (control and test), the test group uses the recommendation tool, and the control group does not or it uses other version than the version proposed in the paper. All the papers that performed an online experiment showed the efficacy of their approach to enhance learning. While online experiments have some advantages faced to offline experiments, like the handling of current (real-time) data, and provide a better planning of the data to be extracted; such kind of evaluation takes more time and cannot guarantee that learning enhancement is caused by the use of the tool. As argued by De Bra [15], this kind of evaluation of turning on or off a feature in a tool does not take in consideration other context elements that could be the cause of an amelioration of learning, it is necessary to separate the evaluation of diverse aspects to pinpoint where a new approach help or fails. For LPRS we could think for instance in providing a recommendation based on a pre-requirement graph and compare with a LPRS that uses different criteria to weight their graph arches, for instance the time limitation. Some common metrics that we can use with an online experiment is the user satisfaction, the system usability and utility, the user opinions and suggestions, and all the metrics used in offline experiments too.

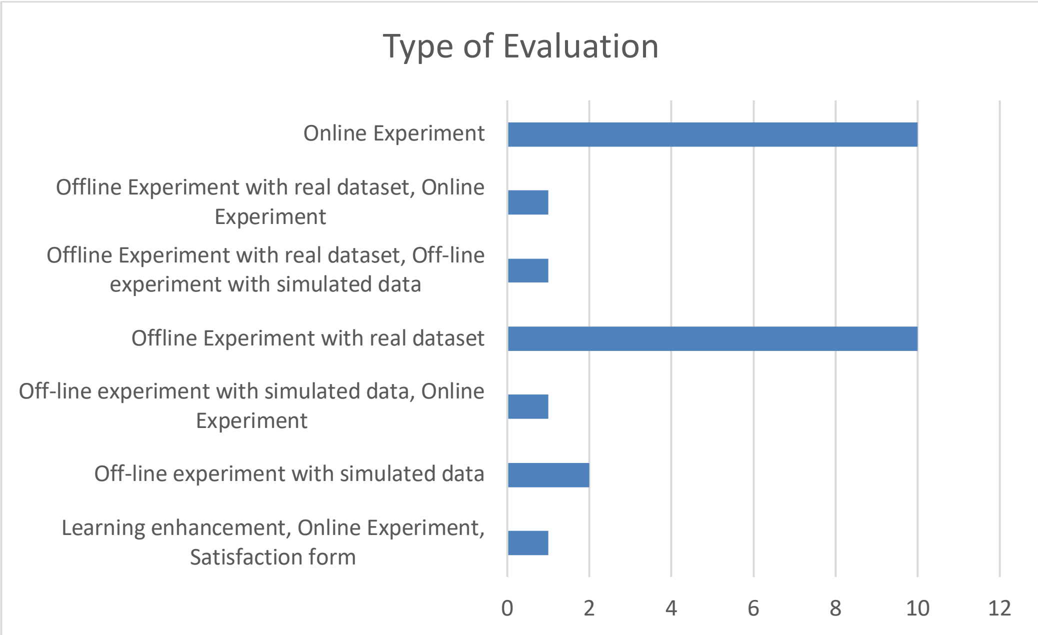

FIGURE 11 - THE KIND OF EVALUATION PERFORMED TO ASSESS THE PROPOSED APPROACH/ALGORITHM

The metric used by 13 of the papers to assess learning was the score in exams, one paper mixt the score with a derivation of it computed by the Item Response Theory (IRT) known as the student ability. The rest of the papers (12 papers) did not assess the impact of their proposition in the enhancement of the student learning. We think one of the reasons for missing the learning evaluation is the paper focusing mainly in presenting a good computational tool that is efficacy in provide recommendations but not looking to the impact such recommendations have to the learning. This result reinforces our criteria of giving more importance to conferences and journals specialized in computer and education, where an educator can question the practical validity of the presented results. The Figure 12 depicts such results.

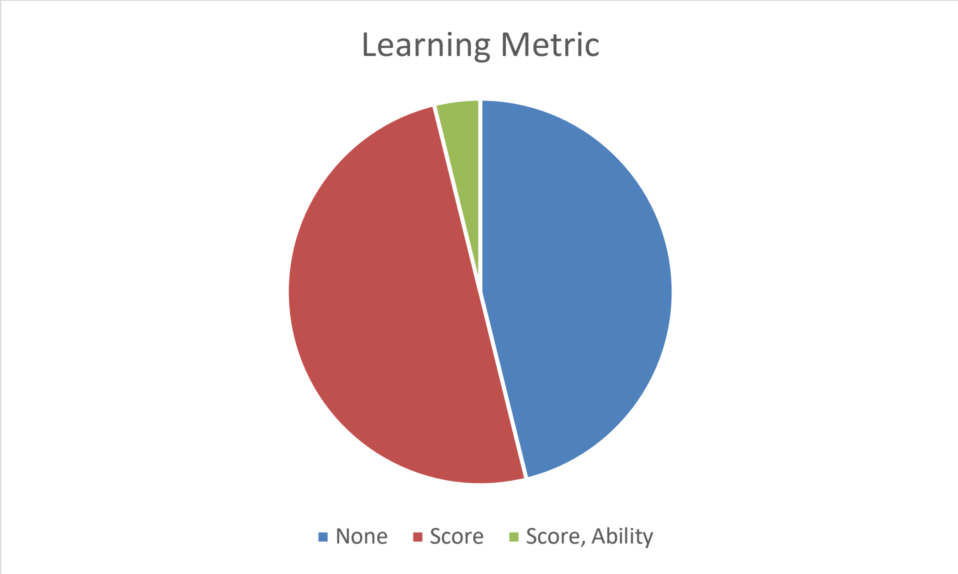

FIGURE 12 - THE METRIC USED TO ASSESS LEARNING

The statistic about the number of metrics used to assess the quality of the generated learning-path are shown in Figure 13. Despite being a generic measure, because we do not ask which metrics are used but the quantity of it, we want to have a general vision about the effort used to evaluate the papers contribution. In our distribution it is possible to see that more than half of the papers assessed their contributions using up to two metrics. To this result we merged both qualitative (satisfaction, opinion, etc.) and quantitative (MAE, Score, Coverage, etc.) metrics. In our observations the most common metrics to assess the quality of the generated learning paths are their own minimization function. This happens because the prediction of a path is often not associated to a rating, what makes it difficult to use metrics as RMSE and MAE, so the papers use an optimization strategy to assure the model quality. In what concerns the learning assessment, as showed before the papers generally use the score.

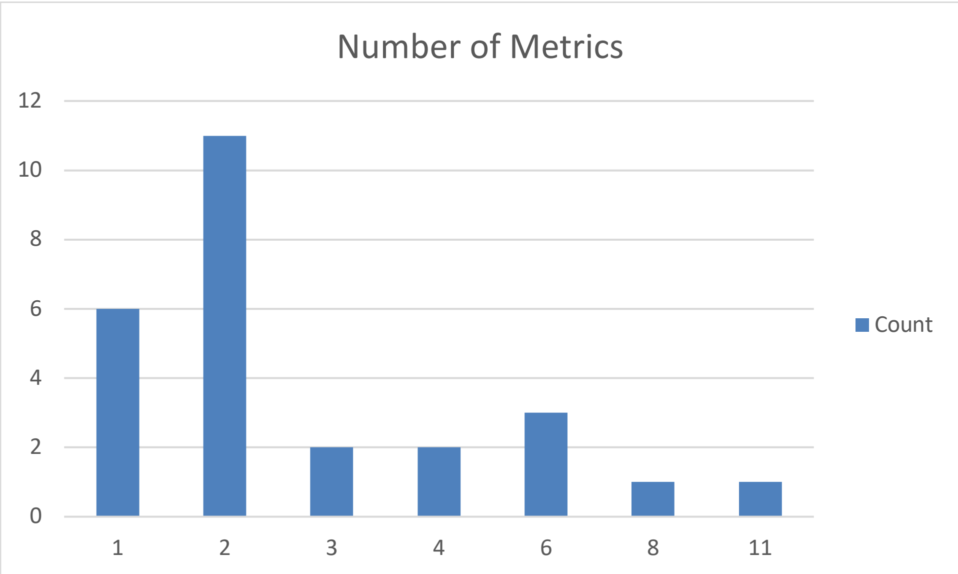

FIGURE 13 - THE DISTRIBUTION OF THE NUMBER OF METRICS USED DURING THE APPROACHES EVALUATION

Generality and reproducibility

We can also identify a clear distinction in what concerns the use of datasets. As we already mentioned none of the papers create a dataset and made it available publicly, other characteristic is the majority of the papers only used one dataset to assess their approach. Which leads us to talk about the need of public available datasets in Technology Enhanced Learning (TEL) domains, specifically for recommendation tasks. There are some efforts to publish TEL datasets as the dataTEL workshop [16], the datasets presented in Kaggle Challenges[9], and big platforms as Open University [17]. However, when reading the papers of this literature mapping, we can see clearly that most of the papers still use their own private dataset. This difficult the comparison of results and the application of the contributions to different scenarios, which can limit the conclusions taken by each author. In such scenario of private datasets we stress the importance of projects like LOLA, which works to make available datasets to make research more reproductible. Another interesting result is the two papers that use 9 datasets were written by the same research group, this shows that the vast majority of the papers evaluate their approaches using up to 4 datasets.

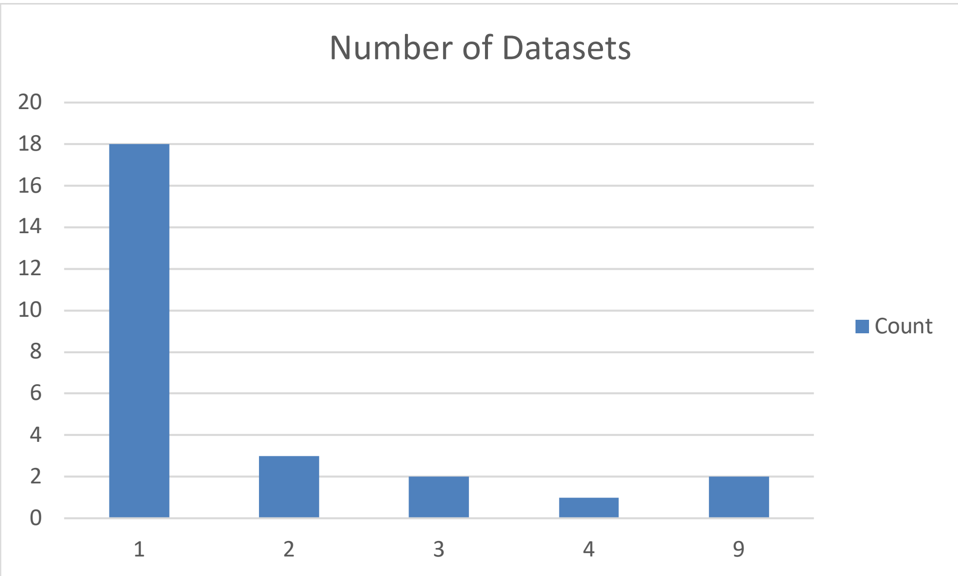

FIGURE 14 - THE NUMBER OF DATASETS USED DURING THE EXPERIMENTS

So, in regard of all the aforementioned results we think in LUNE as a platform that presents many considerable advances to the state-of-the-art. First of all, the platform will use many different datasets, the platform will need a to recommend looking beyond only accuracy, that means the need to combine some different recommendations strategies. The platform will also need to be assessed in a way that the user engagement and trust could be evaluated. Finally, we expected to find more examples of approaches that recommend by considering the user engagement and trust but we did not find any papers dealing with such challenges.

A list containing the papers with individual data and the presented statistics can be found in the link: https://1drv.ms/x/s!AugGORD5_DZWget3M1WvYu9pGeTIMA?e=PQOedk

5. Conclusions and Future Directions for LUNE

[9] https://www.kaggle.com/datasets

This literature survey presented the methodology and results of the systematic mapping performed in the literature at the beginning of the LUNE platform development. The presented statistics showed that the most common approach when building a LPRS **is to look at the content of the learning resources to build a pre-requirement graph which will be used to compute a minimum or maximum weight path to achieve a learning objective.** However other important strategies are the use of a genetic algorithm to achieve an optimized learning path. Authors are still concerned about the changing on the student's level of knowledge during the use of the learning resources, the time the student has available, among other context variable that can impact the use of the LPRS.

Despite of our findings, we conclude that the specificities of LUNE will demand the creation of solutions to deal with the user engagement and trust that were not discussed in any of the found papers.

Some other specific challenges are presented while planning the solution for LUNE platform. Such challenges are related to the nature of the data available to the project, and to the specific objectives. Which leads us to conclude that LUNE demands a novel approach for LPRS. Some of the specific challenges that prevents the use of an off-the-shelf solution are:

- None of the presented papers treated the integration of multiple data and multiple user profiles sources as LUNE demand. Some clues of data integration can be imported directly from the METAL and LOLA projects.
- None of the approaches worked with session-based or few amounts of data situations, which will be the case for LUNE because some of the platforms to be integrated in the solution do not present identified users.
- If LUNE will treat formal learning scenarios, we need to think about the optimization to different learning objectives, since the platform will be used by teachers who want to select resources to formal learning but also by lifelong learners who are engaged on informal learning scenarios.
- The evaluation of the learning also poses a challenge on LUNE platform because in informal learning scenarios the users are not concerned about evaluate the knowledge acquisition, which is essential to validate the efficacy of our platform.
- None of the papers treated specifically the questions about transparency and trustworthy in recommendations.

On the other hand, we cite some of the strategies we can import to structure a good recommendation engine to LUNE:

- The use of a graph to model the items and their pre-requirement relations.
- The use of an optimization method based on the current user learning goal (if we are treating a formal learning scenario or an informal scenario where the user explicated the learning goal). This adaptation of optimization method will help to find the path that maximizes the probability to accomplish the current user learning goal.
- The use of a heuristic to find alternative paths in the graph, that way we can provide an explanation of the reason such path is being recommended.
- The use of the items' content to model the dependency relations between items.
- The performance of an online experiment to collect the user satisfaction with the recommendations and explanations.

The next steps towards the development of the platform should consider all these challenges and strategies to model an architecture that will integrate the multiple sources of data. Once the data sources are integrated, we will need to propose a strategy of recommendation that can be adapted among the ones presented on our review or it can be a totally new way to perform such recommendations, the choice here will depend on the many constraints we will find on the data available to the project. Then the last step we will need to perform the evaluation of the recommendations' accuracy but also effectiveness in explaining the recommendations and ameliorating trust of the users while enhancing engagement.

Finally, I recommend three interesting papers discovered during this review to those who wants to get more knowledge about the development process of a good LPRS. The papers are listed as references [18], [19], [20].

TAKE HOME MESSAGE

- This survey was conducted to depict the state-of-the-art in the field of learning path recommendations.
- We did not find any paper that treats at least one of the challenges of engaging the users, treating their trust, and explaining the recommendations.
- Such characteristics are important because of the LUNE context and our users.
- Our proposition needs to advance the state-of-the-art in such aspects.
- Next step is related to build an architecture that gives accurate recommendations.
- Further we enhance our engine to be trustworthy and transparent.

BIBLIOGRAPHY

[1] P. Brusilovsky, "Adaptive Hypermedia," *User Model. User-adapt. Interact.*, vol. 11, no. 1–2, pp. 87–110, 2001, doi: 10.1023/A:1011143116306.

[2] S. P. Marsh, "Formalising trust as a computational concept," University of Stirling, 1994.

[3] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.

[4] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 1st ed. USA: Cambridge University Press, 2010.

[5] K. Kapoor, V. Kumar, L. Terveen, J. A. Konstan, and P. Schrater, "'I Like to Explore Sometimes': Adapting to Dynamic User Novelty Preferences," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 19–26, doi: 10.1145/2792838.2800172.

[6] P. Massa and P. Avesani, "Trust-Aware Recommender Systems," in *Proceedings of the 2007 ACM Conference on Recommender Systems*, 2007, pp. 17–24, doi: 10.1145/1297231.1297235.

[7] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Model. User-adapt. Interact.*, vol. 12, no. 4, pp. 331–370, 2002, doi: 10.1023/A:1021240730564.

[8] A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, and R. R. Shah, "Learning path personalization and recommendation methods: A survey of the state-of-the-art," *Expert Syst. Appl.*, vol. 159, p. 113596, 2020, doi: https://doi.org/10.1016/j.eswa.2020.113596.

[9] D. Jannach and G. Adomavicius, "Recommendations with a Purpose," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 7–10, doi: 10.1145/2959100.2959186.

[10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015, doi: http://dx.doi.org/10.1016/j.infsof.2015.03.007.

[11] A. H. Nabizadeh, A. M. Jorge, and J. P. Leal, "Estimating time and score uncertainty in generating successful learning paths under time constraints," *Expert Syst.*, vol. 36, no. 2, p. e12351, 2019,

doi: https://doi.org/10.1111/exsy.12351.
[12] M. Birjali, A. Beni-Hssane, and M. Erritali, “A novel adaptive e-learning model based on Big Data by using competence-based knowledge and social learner activities,” *Appl. Soft Comput.*, vol. 69, pp. 14–32, 2018, doi: https://doi.org/10.1016/j.asoc.2018.04.030.
[13] D. Cai, Y. Zhang, and B. Dai, “Learning Path Recommendation Based on Knowledge Tracing Model and Reinforcement Learning,” in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, Dec. 2019, pp. 1881–1885, doi: 10.1109/ICCC47050.2019.9064104.
[14] D. Wang, D. Xu, D. Yu, and G. Xu, “Time-aware sequence model for next-item recommendation,” *Appl. Intell.*, 2020, doi: 10.1007/s10489-020-01820-2.
[15] P. De Bra, “Evaluating Adaptive Systems and Applications is often Nonsense.,” 2016.
[16] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, R. Vuorikari, and E. Duval, “Dataset-Driven Research for Improving Recommender Systems for Learning,” in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 2011, pp. 44–53, doi: 10.1145/2090116.2090122.
[17] J. Kuzilek, M. Hlosta, and Z. Zdrahal, “Open University Learning Analytics dataset,” *Sci. Data*, vol. 4, no. 1, p. 170171, Dec. 2017, doi: 10.1038/sdata.2017.171.
[18] A. H. Nabizadeh, D. Gonçalves, S. Gama, J. Jorge, and H. N. Rafsanjani, “Adaptive learning path recommender approach using auxiliary learning objects,” *Comput. Educ.*, vol. 147, p. 103777, 2020, doi: https://doi.org/10.1016/j.compedu.2019.103777.
[19] P. Dwivedi, V. Kant, and K. K. Bharadwaj, “Learning path recommendation based on modified variable length genetic algorithm,” *Educ. Inf. Technol.*, vol. 23, no. 2, pp. 819–836, Mar. 2018, doi: 10.1007/s10639-017-9637-7.
[20] C. Su, “Designing and Developing a Novel Hybrid Adaptive Learning Path Recommendation System (ALPRS) for Gamification Mathematics Geometry Course,” *EURASIA J. Math. Sci. Technol. Educ.*, vol. 13, no. 6, Apr. 2017, doi: 10.12973/eurasia.2017.01225a.